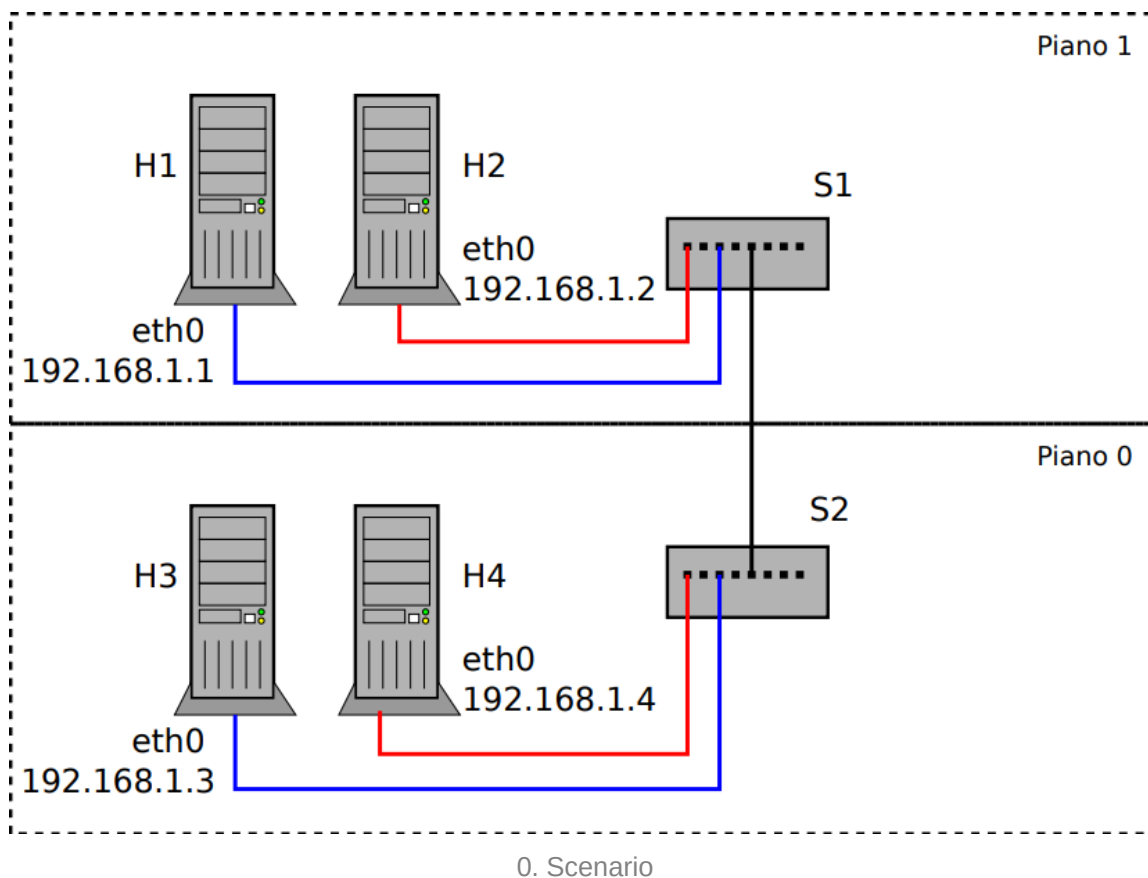


# Esercizio del 15/03/2022

Lo scenario è il seguente



## Obiettivo

Vogliamo realizzare due VLAN sulle quali i PC appartenenti ad esse possano comunicare.

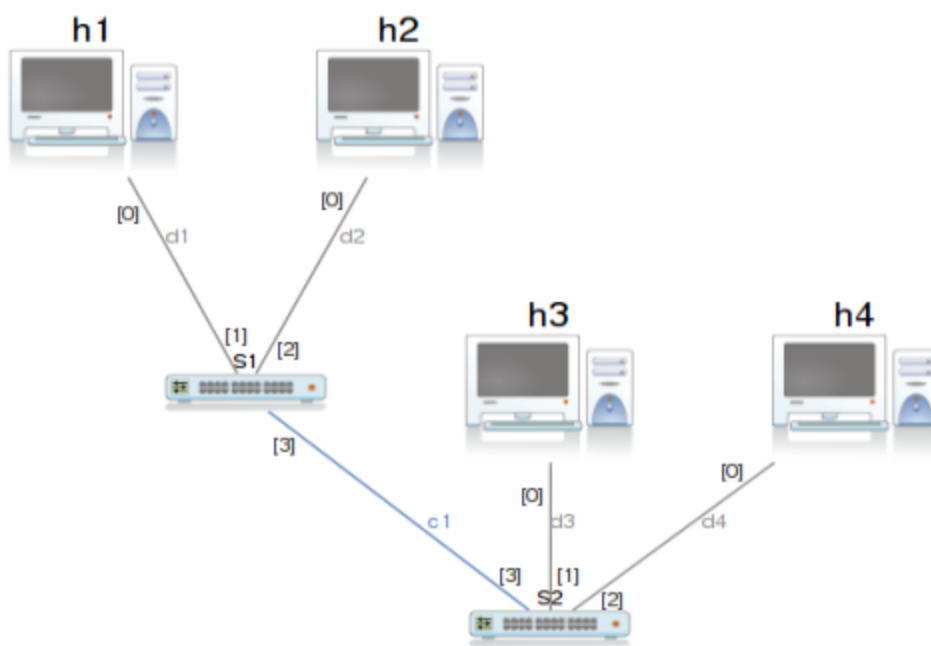
In particolare andremo a creare una vlan che comprende gli host `h1, h3` (VLAN 10) e `h2, h4` (VLAN 20).

## Svolgimento

### 1. Creazione rete “fisica”

Nello svolgimento descritto in questo documento si è fatta la seguente configurazione:

- **h1** è collegato allo switch **S1** sulla porta **1**
- **h2** è collegato allo switch **S1** sulla porta **2**
- **h3** è collegato allo switch **S2** sulla porta **1**
- **h4** è collegato allo switch **S2** sulla porta **2**
- **S1** e **S2** sono collegati da un **cavo cross** sulla porta **3** di entrambi.



1. Schema di riferimento iniziale

## 2. Indirizzamento

In questa esperienza è richiesta la configurazione base dell'interfaccia di rete su ciascuno degli host, ai quali è assegnato l'IP `192.168.1.<n>` sull'interfaccia `eth0`, dove `<n>` è il numero nel nome dell'host (`h1`, `<n> = 1`).

Per fare ciò bisogna avviare i dispositivi e, per ognuno, recarsi nel file

`/etc/network/interfaces` e configurare quanto richiesto:

```
auto eth0
iface eth0 static inet
address 192.168.1.<n>
```

Dopo aver salvato il contenuto del file, eseguire il comando `ifup eth0` per abilitare l'interfaccia con la nuova configurazione.

Una volta fatto ciò noteremo che ciascun dispositivo è raggiungibile da tutti gli altri: questo perché la rete condivide un unico **dominio di broadcast**.

### 3. Creazione VLAN

Poiché vogliamo fare in modo che il dominio di broadcast non sia condiviso da tutti gli host, è necessario frammentare “virtualmente” la nostra rete, introducendo delle VLAN.

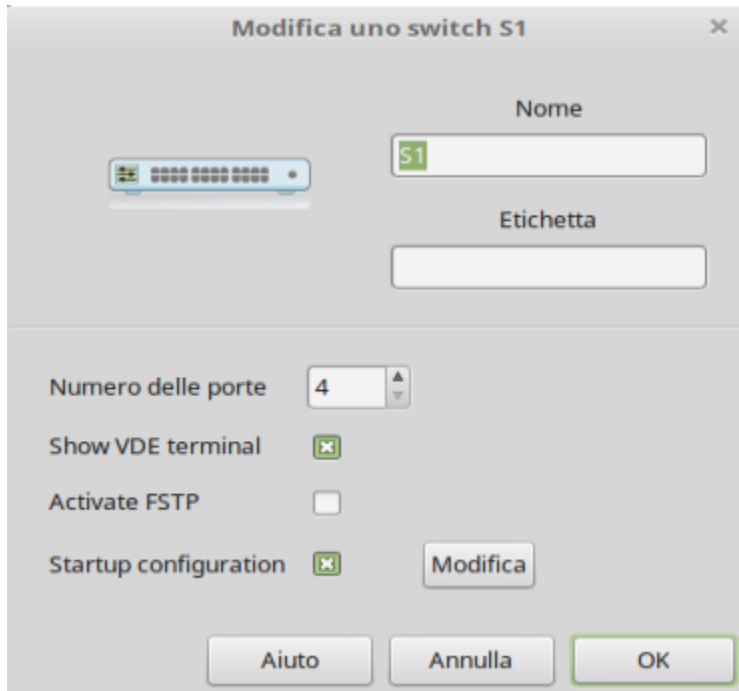
Tutte le porte dello switch, ora, sono **access ports**, ovvero porte che sono configurate come “fisicamente” appartenenti ad una determinata VLAN (in questo caso la VLAN 0, che è quella di default), pertanto lo switch presume che tutto il traffico che viaggia su queste porte appartenga alla VLAN 0.

Se ci rechiamo su uno dei due switch, infatti, eseguendo il comando `vlan/print` noteremo che l'unica VLAN configurata è la `VLAN 0000`, per l'appunto.

Notiamo che tale VLAN ha il flag `tagged` settato a 0: questo significa che su questa VLAN possono transitare pacchetti **taggati e non, poiché è una access port**.

```
vde$ vlan/print
0000 DATA END WITH '.'
VLAN 0000
-- Port 0001 tagged=0 active=1 status=Forwarding
-- Port 0002 tagged=0 active=1 status=Forwarding
-- Port 0003 tagged=0 active=1 status=Forwarding
-- Port 0004 tagged=0 active=1 status=Forwarding
.
1000 Success
```

Arrestiamo tutti i dispositivi e rechiamoci su uno dei due switch per iniziare la configurazione delle VLAN.



3. Finestra di dialogo dello switch: è importante abbia questi flag settati per poter configurare e/o vedere la console dello switch stesso.

Clicchiamo su “Modifica” e iniziamo inserendo:

```
vlan/create 10  
vlan/create 20
```

Questi due comandi ci permettono di creare due VLAN: la 10 e la 20, rispettivamente la VLAN blu e rossa (figura 1).

A questo punto bisogna configurare lo switch in modo tale da assegnare alla porta 1 la VLAN 10 e alla porta 2 la 20 (poiché questi due PC dovranno stare su due domini di broadcast separati).

Inoltre aggiungiamo un **trunk** sulla porta 3: tutti i pacchetti che viaggiano sulla porta 3 **devono essere taggati**. In particolare sul trunk possono viaggiare sia la VLAN 10 che la 20.

```
port/setvlan 1 10 # assegno una porta ad una vlan (porta 1 - vlan 10)  
port/setvlan 2 20 # (porta 2 - vlan 20)  
  
vlan/addport 10 3 # aggiungo sul trunk in porta 3 la vlan 10  
vlan/addport 20 3 # aggiungo sul trunk in porta 3 la vlan 20
```

Pertanto la configurazione completa sarà

```
vlan/create 10
vlan/create 20

port/setvlan 1 10 # assegno una porta ad una vlan (porta 1 - vlan 10)
port/setvlan 2 20 # (porta 2 - vlan 20)

vlan/addport 10 3 # aggiungo sul trunk in porta 3 la vlan 10
vlan/addport 20 3 # aggiungo sul trunk in porta 3 la vlan 20
```

Copiamo la stessa configurazione sullo switch `s2`, poiché sulla porta 1 (risp. 2) dello switch 2 ci sarà `h3` (risp. `h4`).

In questo modo metteremo:

- `h1`, `h3` ⇒ VLAN 10
- `h2`, `h4` ⇒ VLAN 20

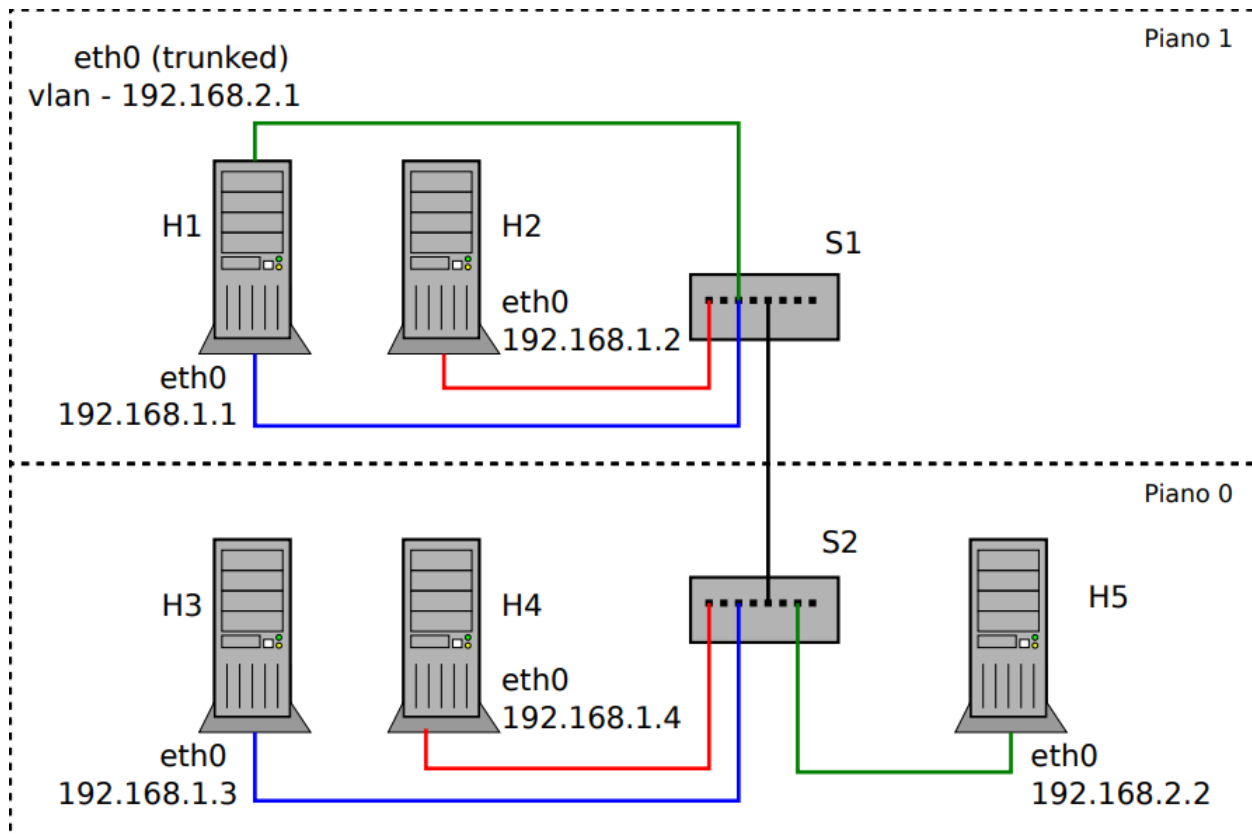
Facendo partire tutto noteremo che:

- se da `h1` eseguo `arping 192.168.1.2` avrò un timeout. Questo comportamento è corretto, dal momento che `h1` e `h2` si trovano su due domini di broadcast diversi (non c'è un router, pertanto l'unico modo che hanno di parlare i due dispositivi è tramite un dominio di broadcast, che è **separato**, per via delle VLAN).
- se da `h1` eseguo `arping 192.168.1.3` avrò una risposta. Comportamento corretto anche in questo caso, dato che `h1` e `h3` condividono lo stesso dominio di broadcast.

**N.B:** in caso di domini di broadcast separati, lo switch **dropa i pacchetti che hanno un dominio di broadcast diverso da quello indicato dalla porta**, pertanto lo sniffing del traffico sull'interfaccia eth0 di `h2` quando si eseguirà il comando `arping` da `h1` non produrrà alcun risultato!

#### 4. Introduzione dell'host `h5`

A questo punto lo scenario viene modificato.



In particolare viene aggiunto l'host `h5`. È richiesto che tale host comunichi tramite VLAN **solo con** `h1`.

Si noti come nello schema sia fatto risaltare il fatto che **non si devono utilizzare altri cavi su** `s1`. **Inoltre si vuole utilizzare la stessa** `eth0` **virtualizzata sulla nuova VLAN (che a questo punto sarà la 30).**

Il problema sarebbe risolvibile facilmente aggiungendo una VLAN in più, configurando un'interfaccia in più su `h1` avente un suo IP e comunicante sulla VLAN condivisa da `h1` e `h5`.

Si vuole risolvere quindi il problema **senza aggiungere ulteriore hardware**, andrà quindi configurata un'interfaccia **virtuale** su `h1` associata esclusivamente alla VLAN 30 (ovvero la nuova VLAN fra `h1` e `h5`).

Facciaml

Per prima cosa configuriamo `h1` per ospitare la nuova interfaccia virtuale:

```
auto eth0.30
iface eth0.30 inet static
address 192.168.2.1
```

Ricordiamo di eseguire `ifup eth0.30`.

Assegnamo un indirizzo anche ad `h5`

```
auto eth0
iface eth0 inet static
address 192.168.2.2
```

Ora non ci resta che configurare gli switch per aggiungere questa nuova VLAN.

È importante notare che a questo punto la porta di collegamento fra `s1` e `h1` diventerà una **porta ibrida** (trunk + access link), dal momento in cui su di essa viaggeranno pacchetti **sia con tag che senza**: la porta 1 agirà come **access port** nel caso il traffico provenga dalla VLAN 10, mentre sarà anche **trunk**, poiché dovrà poter far trasmettere sia frame taggati VLAN 10 che frame taggati VLAN 30.

A questo punto, su `s1` configuriamo il **trunk**:

```
vlan/create 30
vlan/addport 30 1 # rendiamo trunk la porta 1 aggiungendovi la vlan 30
vlan/addport 30 3 # al trunk preesistente aggiungiamo la vlan 30
```

Mentre su `s2` aggiungeremo banalmente una porta **access**

```
vlan/create 30
port/setvlan 4 30 # aggiungiamo l'access port 4 e la mettiamo sulla vlan 30
vlan/addport 30 3 # aggiungiamo al trunk preesistente la vlan 30
```

Le configurazioni finali degli switch saranno:

`S1`

```
vlan/create 10
vlan/create 20
```

```
vlan/create 30

port/setvlan 1 10 # assegno una porta ad una vlan (porta 1 - vlan 10)
port/setvlan 2 20 # (porta 2 - vlan 20)

vlan/addport 10 3 # aggiungo sul trunk in porta 3 la vlan 10
vlan/addport 20 3 # aggiungo sul trunk in porta 3 la vlan 20
vlan/addport 30 1 # rendiamo trunk la porta 1 aggiungendovi la vlan 30
vlan/addport 30 3 # al trunk preesistente aggiungiamo la vlan 30

# la coesistenza di port/setvlan 1 10 e vlan/addport 30 1 rende la porta 1
# una porta ibrida!
```

S2

```
vlan/create 10
vlan/create 20
vlan/create 30

port/setvlan 1 10
port/setvlan 2 20
port/setvlan 4 30 # aggiungiamo l'access port 4 e la mettiamo sulla vlan 30

vlan/addport 10 3
vlan/addport 20 3
vlan/addport 30 3 # aggiungiamo al trunk preesistente la vlan 30
```

Emiliano Maccaferri